

# Walking and Steering Control for a 3D Biped Robot Considering Ground Contact and Stability

Ting, Wang<sup>a,\*</sup>, Christine, Chevallereau<sup>a</sup>, Carlos F., Rengifo<sup>b</sup>

<sup>a</sup>*IRCCyN, CNRS, Ecole Centrale de Nantes, 1 Rue de la Noë, 44321, Nantes, cedex 03,  
France*

<sup>b</sup>*Universidad del Cauca, Calle 5 No 4-70, Popayán, Colombia*

---

## 1. Introduction

The walking of a bipedal robot is composed of different phases such as single support (SS) with flat foot, SS with foot rotation around the metatarsal axis and double support. During every walking phase there is a dynamic model and a contact with the ground. Thus, an appropriated control law has to be implemented for each phase. Generally, the type of contact with the ground is imposed but it is obtained only if the conditions on ground reaction forces, friction cone and zero moment point (ZMP) are satisfied. The ZMP, firstly introduced by Vukobratovic [1], is the most important factor in achieving stable walking of an humanoid robot. The ZMP is defined as a point where the resulting horizontal reaction moment generated by the ground reaction forces equals to zero. If this point is inside of the support polygon, the robot will not rotate around the edges of its stance foot so it remains flat on the ground. The ZMP can be used to measure instantaneous balance at each time step but it does not contain notion of stability in the future [2].

In earlier studies many bipedal robots adopted a control strategy in which the desired trajectories of joint angles are firstly designed based on the ZMP condition and after that a feedback control tracks the desired trajectories [1]. However, contact conditions could not be satisfied in presence of perturbations.

---

\*Corresponding author: Ting.Wang@irccyn.ec-nantes.fr

Most of the recent research on ZMP control can be divided into two general approaches. One method is the periodical replaying of trajectories for the joint motions recorded in advance, like [1], which are then applied to the real robot with a little computed on-line modification [3], [4], [5]. This strategy explicitly divides the problem into subproblems of planning and control. Another method generates a joint-motion in real time, feeding back the present state of the system to be in accordance with the pre-provided goal of the motion, where planning and control are managed in a unified way [6], [7], [8], [9], [10], [11]. Two of the more famous users of these methods are Honda Robot Asimo [3] and Kawada's humanoid HRP-2 [12], [13].

Almost all of the computed on-line walking controllers are applied to compensate the ZMP error. However, the general weakness of the previous methods is that they require considerable experimental hand tuning and the proposed methods are poorly documented. As a preliminary study of our robot, the control law proposed in this paper belongs to the first family of methods. The reference trajectory of the joint angles and ZMP have been computed off-line in [14]. Our first objective is to propose a new control law to satisfy the constraint of contact and to study the stability of walking, that is, the convergence towards a periodical walking. The computed on-line control modification is not based on experimental tests but on a rigorous stability study.

The proposed control law consists of ZMP controller, swing ankle rotation controller and partial joint angles controller. In the first one, the positions of ZMP in the horizontal plane are regulated to the desired ZMP. This controller creates 2 constraints on joint accelerations and by consequence not all of the joints can track the desired motion. If the obtained joint motion differs from the desired one, a flat-foot impact for the swing foot cannot be assured. In such a case, the stance foot will not be in "flat-foot-contact" with the sole at the beginning of the next single-support phase. In order to avoid this problem a swing ankle rotation controller is used. In this one, the pitch and roll angle of the swing foot are controlled in order to be equal to their desired values. Like the ZMP controller, the swing ankle rotation controller creates 2 constraints on

joint accelerations. Consequently, for a 3D robot with  $n$  actuated joints,  $n - 4$  controlled outputs can be chosen. For simplicity, these  $n - 4$  controlled outputs, which are noted as  $u$ , are defined as a linear combination of the  $n$  actuated joints. The partial joint angles controller is applied to make  $u$  track its desired value  $u^d$ . As a result, the three controllers can offer  $n$  dynamic equations to resolve the acceleration of  $n$  actuated joints, then the input torque can be obtained by the inverse dynamics.

It is important to note that in the partial joint angles controller the choice of controlled outputs  $u$  depends on the stability analysis of the walking gait under closed-loop control, which is evaluated with the linearisation of the restricted Poincaré map of the hybrid zero dynamics [15]. In our previous studies [16] and [17], it has been showed through examples, that the choice of controlled outputs will influence the walking stability and with an appropriate selection of these variables a stable walking cycle can be obtained.

In short, the ZMP and swing ankle controllers are used to ensure the stable condition of supported foot and transfered foot respectively. The partial joint angles controller is used to track the reference trajectory and satisfy the stable condition of the overall control law. This control strategy is original for the bipedal robot with feet. Compared to the general control law in recent studies, our control law has some advantages as follows. The proposed method can be viewed as a computed on-line modification of the reference trajectory in order to ensure the satisfaction of the contact constraint. The effect of this modification on walking stability is based on a rigorous stability analysis, and not by testing on the robot which requires considerable experimental hand tuning.

With the proposed control law, the robot can achieve an asymptotically stable and periodic walking along a straight line. The second objective of this paper is adjusting the net yaw rotation of the robot over a step in order to steer the robot along paths with mild curvature. An event-based (or stride-to-stride) feedback controller is appended to distribute set point commands to all the actuated joints in order to achieve a desired amount of turning, as opposed to the continuous corrections used in [18]. This work is an extension from the previous

study for a 3D underactuated bipedal robot [19] and an interesting feature is the ability to control the robot's motion along paths with mild curvature using only a single predefined periodic motion.

This paper is organized as follows. Section 2 is a description of the biped robot and its dynamic models during different walking phases. Section 3 presents the reference trajectory of the joint angles and ZMP. Section 4 introduces the principle of control law including ZMP controller, swing ankle rotation controller and partial joints controller, and it also introduces the modification of the reference motion after the impact phase to create a continuous and cyclic desired trajectory. In Section 5, the hybrid zero dynamic is explained and stability analysis based on Poincaré method is proposed. Section 6 gives the comparison of our method with a classical control law for the same robot walking on the rigid ground model and the soft ground model respectively. In Section 7, a supplemental event-based control law used to regulate the walking direction of the robot is presented. Some examples of walking direction control are given. Concluding remarks are made in Section 8.

## 2. Model

### 2.1. Biped Model

As shown in Fig. 1, the 3D robot is comprised of a torso and two identical legs that are independently actuated and terminated with flat-feet. Each hip and ankle consists of a revolute joint with 3 DOF and the knee is a 1 DOF revolute joint, and each DOF is independently actuated. As a result, it has 14 DOF in the single support phase and 14 actuators. The vector  $q = [q_1, \dots, q_{14}]^T$  contains the relative joint positions of the biped, and the torques are grouped into a torque vector,  $\Gamma = [\Gamma_1, \dots, \Gamma_{14}]^T$ , where  $q_j$ ,  $j \in (1, \dots, 14)$  corresponds to the rotation about  $z_j$ . It's worth noting that the coordinate system  $x_0y_0z_0$  is fixed on the support foot while the coordinate system  $x_sy_s z_s$  is fixed on the ground and it is called the absolute coordinate system. The origins of them are the same only during the first walking step.

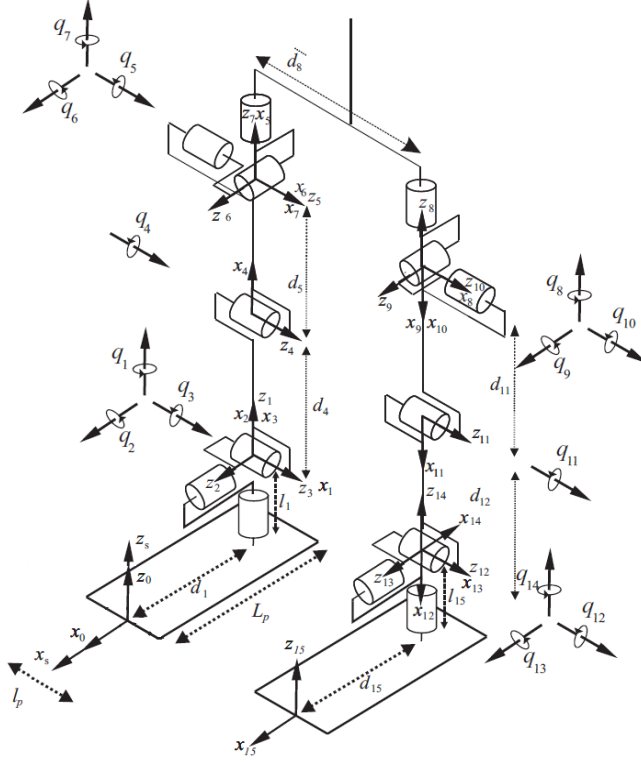


Figure 1: Biped Model

The length and mass of each body are given in Table 1 and Table 2. All links are assumed to be massive and rigid. The inertia of the links is also taken into account, and other parameters of the robot in detail are shown in [14].

Table 1: length parameters(m)

$d_1, d_{15}$	$d_4, d_{12}$	$d_5, d_{11}$	$d_8$	$l_1, l_{15}$	$l_p$	$L_p$
0.185	0.392	0.392	0.190	0.05925	0.08	0.207

Table 2: mass parameters

<i>body</i>	<i>foot</i>	<i>shin</i>	<i>thigh</i>	<i>torso</i>
<i>kg</i>	0.678	2.188	5.025	24.97

## 2.2. Dynamic model

The walking gait consists of single support phases where the stance foot is flat on the ground separated by impacts, that is, instantaneous double support phases where leg exchange takes place. The dynamic models for every walking phase are derived here by assuming support on right leg, which is noted as leg 1. They are described as follows, and the models for support on leg 2 can be written in a similar way.

### 2.2.1. Dynamic model of single support phase

The Newton Euler algorithm presented in [20] is used to calculate the dynamic model of 3D biped robot during the single support phase [14]. The actuator torques and the ground reaction (forces and torques) to the stance foot can be computed as soon as the joint position, velocity and acceleration  $q, \dot{q}, \ddot{q}$  are known, and they are linear functions of the acceleration  $\ddot{q}$ . Consequently, the dynamic model is represented as:

$$\begin{bmatrix} F \\ M \\ \Gamma \end{bmatrix} = NE(q, \dot{q}, \ddot{q}) = NE_1(q)\ddot{q} + NE_2(q, \dot{q}) \quad (1)$$

$NE_1(q)$  being the positive definite inertia matrix of the robot and  $NE_2(q, \dot{q})$  a vector containing the gravitational, Coriolis and centrifugal forces.  $F$  and  $M$  compose the wrench vector exerted by the ground on the stance foot. Both quantities are expressed in the reference frame  $x_0y_0z_0$  and they will be used to calculate the position of ZMP. The calculation of the torque vector  $\Gamma$  will be used in the control law.

### 2.2.2. ZMP dynamics

When leg-1 is on the ground, a ground reaction force and moment can be expressed about any point in the support plane. If a point  $P$  is in the plane of the foot-1 (see Fig. 2), the ground reaction moment  $M_p$  of  $P$  can be calculated by:

$$M_p = M - \vec{OP} \times F \quad (2)$$

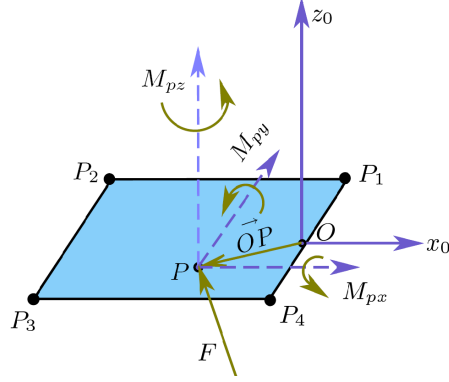


Figure 2: The support foot

where  $\vec{OP}$  is the distance vector of the point  $P$  in the reference frame  $x_0y_0z_0$ . Considering  $F, M$  are functions of  $q, \dot{q}, \ddot{q}$  and linear functions of the acceleration  $\ddot{q}$  (see (1)), the components of  $M_p$  along the axes  $x_0$  and  $y_0$  can be written as:

$$\begin{bmatrix} M_{px} \\ M_{py} \end{bmatrix} = f(q, \dot{q}, \ddot{q}, \vec{OP}) = W\ddot{q} + H \quad (3)$$

where the function  $f()$  in (3) depends on (1) and (2), and  $W, H$  can be obtained by:

$$H = f(q, \dot{q}, \ddot{q} = \mathbf{0}_{14 \times 1}, \vec{OP}), \quad (4)$$

and

$$W(:, i) = f(q, \dot{q}, \ddot{q} = \mathbf{e}_i, \vec{OP}) - H \quad (5)$$

$\mathbf{e}_i$  being the column  $i$  of an identity matrix of dimension 14. According to the definition of zero moment point (ZMP), if  $P$  is ZMP, the horizontal ground reaction moment of the ground reaction force about this point equals to zero. Thus (3) can be used to constrain the ZMP which is such that:

$$M_{px} = M_{py} = 0 \quad (6)$$

to be a desired point  $P$  which is time-varying and depends on the precomputed reference trajectory [1].

### 2.2.3. Dynamic model of impact phase

An impact exists at the end of the single support phase. After that, the legs swap their roles from one step to the next. Considering the robot is symmetric, we only study a single step to deduce the complete behavior of the robot over a sequence of steps on alternating legs. During the impact, the biped's configuration variables do not change, but the generalized velocities undergo a jump. As shown in [15], this jump is linear with respect to the joint velocity before the impact. The moments just before and after impact are denoted by  $-$  and  $+$  respectively, the impact model can be written as (see [14]):

$$\begin{cases} q^+ = E \cdot q^- \\ \dot{q}^+ = E \cdot I(q^-) \cdot \dot{q}^- \end{cases} \quad (7)$$

where  $E$  is the permutation matrix describing leg's exchange and  $I(q^-)$  represents the linear jump of the joint velocity after impact.

### 3. The reference trajectory

Starting from a reference trajectory of joint angles described as function of time  $q^d(t)$  which has been calculated off-line through the minimization of the energy consumption in [14], a new parametrization is proposed. Here and in the following subsections the superscript  $d$  denotes the desired value. Function  $q^d(t)$  is parametrized by a quantity depending on the state of the robot that is strictly monotonic like time  $t$  during a complete walking phase of one step. By using this method, only the kinematic evolution of the robot's state is regulated but not its temporal evolution. This means the control law is defined to follow a joint path but not a joint motion. It was used successfully for the walking of bipeds with point feet [21], [22], [23], [16] and the walking of planar biped with foot rotation [24]. Especially in [24], this method allows the simultaneous control of the joint path and the ZMP. Therefore, it is still used for the 3D biped robot with flat feet. The parametrized reference trajectory is given as follows.

In a forward walking motion, the  $x$ -coordinate of the hip increases monotonically. Hence, if the virtual stance leg is defined by the line that connects the



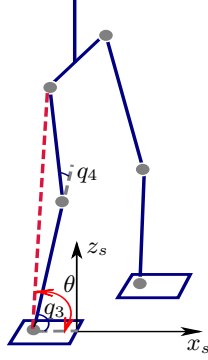


Figure 3: The definition of  $\theta$ .

stance foot to the stance hip, the angle of this leg in the sagittal plane, denoted by  $\theta$ , is monotonic and it can replace the time  $t$  to parametrize  $q^d$  and  $ZMP^d$ . As shown in Fig. 3, the shin and the thigh have the same length, thus  $\theta$  can be computed by:

$$\theta = q_3 + q_4/2. \quad (8)$$

Then the reference path written as  $h^d(\theta)$  is such that:

$$\begin{cases} q^d(t) = h^d(\theta) \\ \dot{q}^d(t) = \frac{\partial h^d(\theta)}{\partial \theta} \dot{\theta} \\ \ddot{q}^d(t) = \frac{\partial^2 h^d(\theta)}{\partial \theta^2} \dot{\theta}^2 + \frac{\partial h^d(\theta)}{\partial \theta} \ddot{\theta} \end{cases} \quad (9)$$

where  $h^d(\theta)$  is designed in order to be compatible with a periodic solution of the biped model. According to a known reference trajectory  $q^d(t)$ ,  $\dot{q}^d(t)$ ,  $\ddot{q}^d(t)$ , in (9)  $h^d(\theta)$ ,  $\frac{\partial h^d(\theta)}{\partial \theta}$  and  $\frac{\partial^2 h^d(\theta)}{\partial \theta^2}$  can be deduced and they are assumed to be known in the following contents. Similarly, for every  $q^d(t)$ , there is a correspond  $\theta$  and  $ZMP^d(t)$ , then the desired ZMP can be described by  $\theta$  and it is written as:  $ZMP^d(\theta)$ .

#### 4. The proposed control law

As shown in Fig. 4, the overall control law consists of ZMP controller, swing ankle rotation controller and partial joint angles controller. The former two

controllers are used to achieve  $ZMP = ZMP^d(\theta)$  and make the swing foot keep flat at impacts respectively, which are helpful to insure the stable contact. The third controller implements the joint path control and it is partial since some constraints on torques are imposed by the former two controllers.

#### 4.1. ZMP controller

The objective of this controller is to have:

$$ZMP = ZMP^d(\theta). \quad (10)$$

ZMP tracking can be quite conservative but insure to avoid unexpected rotation of the stance foot. Another possibility is to limit the minimal and maximal values of ZMP [25]. According to the definition of ZMP, if  $P$  in Fig. 2 is ZMP, (3) can be rewritten as:

$$W(q, \dot{q})\ddot{q} + H(q, \dot{q}) = \mathbf{0}_{2 \times 1} \quad (11)$$

where  $W$  and  $H$  are recalculated by (4) and (5) with  $\vec{OP} = ZMP^d(\theta)$ . This equation clearly shows that the acceleration  $\ddot{q}$  must satisfy two constraints to produce  $ZMP = ZMP^d(\theta)$ .

#### 4.2. Swing ankle rotation controller

With the reference trajectory  $q^d, \dot{q}^d, \ddot{q}^d$ , the sole of the transferred foot parallels to the ground at each impact phase. As stated in the Introduction, this controller is necessary because not all of the joint angles  $q$  are controlled. We propose to control the orientation of the swing foot in the frontal and sagittal plane such that the robot will touch the ground with flat foot and to insure that the next impact will occur in a good way.

As shown in Fig. 1, the angle  $q_{14}$  is the rotation angle of the 14<sup>th</sup> joint around the axes  $z_{14}$ . It only affects the yaw motion of the swing foot, so only the previous 13 joints are constrained. Define  ${}^sR_{13}$ , the  $3 \times 3$  orientation matrix from the coordinate system  $x_s y_s z_s$  to  $x_{13} y_{13} z_{13}$ :

$${}^sR_{13} = \begin{bmatrix} s & n & a \end{bmatrix}, \quad (12)$$

Figure 4: Block diagram of the control system.

where  $s_{3 \times 1}$ ,  $n_{3 \times 1}$  and  $a_{3 \times 1}$  are unitary vectors defining  $x_{13}y_{13}z_{13}$  in frame  ${}^sR$  respectively, and they only depend on the first thirteen joint angles of  $q$ , exactly,  $q_1, q_2, \dots, q_{13}$ . Using the reference trajectory  $q^d$ , we define  ${}^sR_{13}^d = [s^d, n^d, a^d]$  to denote the desired orientation of transformation matrix  ${}^sR_{13}$ . If there exists tracking error between  $q$  and  $q^d$ , there will be small error in the orientation of the swing ankle which can be defined as  $\xi$  [26]:

$$\xi = \frac{1}{2}(s \times s^d + n \times n^d + a \times a^d). \quad (13)$$

The derivatives of the orientation error are :

$$\begin{cases} \dot{\xi} = {}^s\omega_{13}^d - {}^s\omega_{13} \\ \ddot{\xi} = {}^s\dot{\omega}_{13}^d - {}^s\dot{\omega}_{13} \end{cases} \quad (14)$$

where  ${}^s\omega_{13}$  and  ${}^s\dot{\omega}_{13}$  are the angular acceleration and velocity of the joint 13 in the absolute coordinate system  $x_sy_s z_s$ .  ${}^sR_{13}$ ,  ${}^s\omega_{13}$  and  ${}^s\dot{\omega}_{13}$  in (12) can be efficiently computed by the forward recursive equations of the general serial robot with  $q_1, \dots, q_{13}$ , their velocity and acceleration [20] (see Appendix Appendix A).

Because the orientation in the frontal and sagittal plane of the swing foot doesn't depend on the rotation components around  $z_s$  axis (see Fig. 1), only the first two components of  $\xi$  have to be regulated to zero:

$$\xi_{xy} = \mathbf{0}_{2 \times 1} \quad (15)$$

Here and in the following subsections the subscript  $xy$  denotes the first two lines of the vector, which correspond to the rotation around  $x_s$  direction and  $y_s$  direction respectively. (15) can be achieved by the PD controller:

$$\ddot{\xi}_{xy} + \frac{K_1}{\varepsilon} \dot{\xi}_{xy} + \frac{K_2}{\varepsilon^2} \xi_{xy} = \mathbf{0}_{2 \times 1}. \quad (16)$$

with  $K_1 > 0$ ,  $K_2 > 0$ , and  $\varepsilon > 0$ . The above equation imposes two constraints on  $q$ ,  $\dot{q}$  and  $\ddot{q}$ . Such constraints read as:

$$W_s(q, \dot{q})\ddot{q} + H_s(q, \dot{q}) = \mathbf{0}_{2 \times 1}. \quad (17)$$

where  $W_s(q, \dot{q})$  denotes all the coefficient terms of  $\ddot{q}$  and  $H_s(q, \dot{q})$  denotes all the terms without  $\ddot{q}$  in (16). The detail is presented in Appendix Appendix B.

#### 4.3. Partial joint angles controller

Besides of the former two controllers, the control law is also designed to track the joint reference path  $q^d = h^d(\theta)$ . However, (11) and (17) have offered 4 restrictions about the states of robot  $q, \dot{q}, \ddot{q}$ , so we can't control  $q = q^d$  directly and we can only choose 10 targets to track with. Thus, how to choose 10 controlled outputs from 14 actuated joints is presented at first.

From the principle of swing ankle rotation controller, we know that the pitch and roll angle of the swing foot are constrained to be their desired values. It means that 2 joint angles can be determined by the other 11 joint angles in  $q_1, q_2, \dots, q_{13}$  according to the definition of  $\xi_{xy}$  in (12) and (13).

Because the swing ankle is actuated by the 12<sup>th</sup> and 13<sup>th</sup> joint (see Fig. 1), so we can suppose that  $q_{12}$  and  $q_{13}$  have been defined according to  $q_1, q_2, \dots, q_{11}$ . Thus there is no need to consider them again in the partial joint angles controller. In addition, section 3 has presented that all the reference trajectory are parametrized by  $\theta$  and  $\theta = q_3 + 0.5q_4$ , so  $q_3$  and  $q_4$  are not independent of each other and  $q_3$  is excluded here. Therefore, the independent joint angles are:

$$Q_r = [q_1, q_2, q_4, \dots, q_{11}, q_{14}]_{11 \times 1}^T \quad (18)$$

$Q_r$  has 11 elements but only 10 controlled outputs can be defined. Since the second joint influences the position of ZMP in the frontal plane directly, in order to distinguish  $q_2$  from other joints in  $Q_r$  and avoid singularity, the controlled joint angles are chosen as:

$$u = Q + M_1 q_2 \quad (19)$$

with

$$Q = [q_1, q_4, \dots, q_{11}, q_{14}]^T. \quad (20)$$

Where  $M_1$  is a  $10 \times 1$  constant matrix to be determined by the stability analysis. The next section will show that the choice of  $M_1$  affects the stability of the robot, in the sense of the convergence toward cyclic motion. The stability condition will be a criteria used to select  $M_1$ .

Next, the objective of this controller is to regulate  $u$  to track its desired value  $u^d(\theta)$ . It is achieved by a PD control:

$$\ddot{u} - \ddot{u}^d + \frac{K_d}{\varepsilon}(\dot{u} - \dot{u}^d) + \frac{K_p}{\varepsilon^2}(u - u^d) = \mathbf{0}_{10 \times 1}, \quad (21)$$

where  $u^d$ ,  $\dot{u}^d$ ,  $\ddot{u}^d$  can be calculated using the definition of  $u$  in (19) and the referenced trajectory  $q^d$ ,  $\dot{q}^d$ ,  $\ddot{q}^d$  in (9). Because the controlled partial joint angles  $u$  is a linear function of the joint angles  $q$  (see (19)), obviously,  $\ddot{u}$  in (21) is a linear function of  $\ddot{q}$ , so (21) can be written as the same format as (11) and (17):

$$W_j(q, \dot{q})_{10 \times 14} \ddot{q} + H_j(q, \dot{q})_{10 \times 1} = \mathbf{0}_{10 \times 1}, \quad (22)$$

where  $W_j(q, \dot{q})$  denotes all the coefficient terms of  $\ddot{q}$  and  $H_j(q, \dot{q})$  denotes all the terms without  $\ddot{q}$  in (21).

It is worth mentioning that if  $u = u^d(\theta)$ , according to (19),  $Q$  is such that:

$$Q = Q^d - M_1(q_2 - q_2^d). \quad (23)$$

(23) implies that this controller can also be regarded as an on-line modification of the joint reference motion, which is like many existent methods. However, the advantage of our controller is that the effect of this on-line modification (or the choice of  $M_1$  in (23)) on stability will be studied.

#### 4.4. Modification of the tracked errors caused by impact

Based on the proposed three controllers, the desired properties respectively are:

$$\begin{cases} y_1 = ZMP - ZMP^d(\theta) = 0 \\ y_2 = \xi_{xy} = 0 \\ y_3 = u - u^d(\theta) = 0 \end{cases} \quad (24)$$

These controllers act only in the single support phases. The original reference trajectory is defined by taking into account the impact phase. The impact condition is a geometric condition. Thus, it can occur with some errors, especially with some errors on  $\theta$  and  $q_2$  because they are not controlled directly as other states in  $u$ . As a result, large errors on the joint position and velocity will be created after impact, that means discontinuity of errors will exist.

In order to avoid the discontinuity, the reference trajectory in (24) are modified stride to stride so that they are compatible with the initial state of the robot at the beginning of each step [16]. It is not necessary to modify  $ZMP^d(\theta)$  because the position of ZMP acts directly at the acceleration level where the discontinuities are acceptable. The new output for the feedback control design is:

$$\begin{cases} y_1 = ZMP - ZMP^d(\theta) = 0 \\ y_{2c} = \xi_{xy} - \xi_{xyc} = 0 \\ y_{3c} = u - u^d(\theta) - u_c = 0 \end{cases} \quad (25)$$

Here  $\xi_{xyc}$  and  $u_c$  are modification terms of the original reference motions. The calculation of them is presented as follows.

The function  $u_c$  is taken to be a four-order continuously differentiable polynomial function of  $\theta$ :

$$u_c = a_0 + a_1\theta + a_2\theta^2 + a_3\theta^3 + a_4\theta^4, \quad (26)$$

where the coefficients  $a_0, \dots, a_4$  are defined such that <sup>1</sup>

$$\begin{cases} u_c(\theta_i) = y_{3i} \\ \frac{\partial u_c}{\partial \theta}(\theta_i) = \frac{\dot{y}_{3i}}{\theta_i} \\ u_c(\theta) \equiv 0, \quad \theta \geq \frac{\theta_i + \theta_f}{2}. \end{cases} \quad (27)$$

where  $y_{3i}$  and  $\dot{y}_{3i}$  denote the initial values of  $y_3$  and  $\dot{y}_3$ , and they are updated at the beginning and held constant throughout the step. Specifically,  $y_{3i} = u_i - u^d(\theta_i)$  and  $\dot{y}_{3i} = \dot{u}_i - \dot{u}^d(\theta_i)$ . Here  $\theta_i$  and  $\theta_f$  are the initial and final value of  $\theta$  from  $q_i$  to  $q_f$  respectively, and they can be calculated by (8).

As shown in Fig. 19, by using  $u_c$  defined in this way the new reference trajectory starting from the current state just after the impact are smoothly joined to the original reference trajectory at the middle of the step. In particular, for any initial error, the initial reference trajectory  $u^d$  is not modified in the second part of the step :  $\theta \geq \frac{\theta_i + \theta_f}{2}$ .

---

<sup>1</sup>The four order polynomial is used for  $\theta_i \leq \theta \leq \frac{\theta_i + \theta_f}{2}$ ; Continuity of position, velocity and acceleration is ensured at  $\theta = \frac{\theta_i + \theta_f}{2}$ .

Similarly, the function of  $\xi_{xyc}$  in (25) is defined as the same format as (26) and all the coefficients are determined using the value of  $y_2$  in (24) and its derivative at the beginning of each step. As a result, these modification terms are supplemented in the PD control expressions (16) and (21):

$$\ddot{\xi}_{xy} - \ddot{\xi}_{xyc} + \frac{K_1}{\varepsilon}(\dot{\xi}_{xy} - \dot{\xi}_{xyc}) + \frac{K_2}{\varepsilon^2}(\xi_{xy} - \xi_{xyc}) = \mathbf{0}_{2 \times 1} \quad (28)$$

$$\ddot{u} - \ddot{u}^d - \ddot{u}_c + \frac{K_d}{\varepsilon}(\dot{u} - \dot{u}^d - \dot{u}_c) + \frac{K_p}{\varepsilon^2}(u - u^d - u_c) = \mathbf{0}_{10 \times 1} \quad (29)$$

#### 4.5. Torque calculation

With the proposed three controllers, 14 equations about  $q, \dot{q}, \ddot{q}$  are obtained by (11), (17) and (22), and they are affine functions of  $\ddot{q}$ . Considering the impact modifications, the latter two controllers (17) and (22) are recreated using (28) and (29) to satisfy (25), and they are also affine functions of  $\ddot{q}$ . Thus, 14 new equations about  $q, \dot{q}, \ddot{q}$  can be combined as:

$$A(q, \dot{q})\ddot{q} + B(q, \dot{q}) = \mathbf{0}_{14 \times 1}, \quad (30)$$

Next, the  $\ddot{q}$  is resolved directly:

$$\ddot{q} = -A^{-1}B \quad (31)$$

Then, the desired torques  $\Gamma$  can be calculated with  $\ddot{q}, \dot{q}$  and  $q$  by the inverse dynamics via Newton Euler algorithm (1).

## 5. Stability analysis

The stability of the control law is defined in the sense of the convergence toward a periodic motion. As proposed in [21], [22], [23], [16] and [15, Chap. 4], the stability of the closed-loop system can be studied in a reduced-dimensional state space based on Hybrid Zero Dynamics (HZD). For a system modeled by ordinary differential equations (in particular, without impact dynamics), the maximal internal dynamics of the system that is compatible with the output being identically zero is called the zero dynamics [27], [28]. HZD is a zero dynamics of the full hybrid model, i.e., the dynamics at impact is included.



Next we will present how to study the stability of the closed-loop system in a reduced-dimensional state space.

### 5.1. Using HZD to obtain reduced-dimensional states

According to the definitions of HZD, the controller with impact modifications (11), (28) and (29) creates the HZD, with which the behavior of the robot is completely defined by the impact map and the swing phase zero dynamic model (25). The stability analysis of walking can be obtained by using the Poincaré method. Different Poincaré section can be considered. Here it is chosen at  $\theta = 0.4\theta_i + 0.6\theta_f$ . At this instant the swing foot has not yet touched the ground, and since  $\theta \geq \frac{\theta_i + \theta_f}{2}$  the values of the controlled variables  $u$  and  $\xi_{xy}$  are not affected by  $u_c$  and  $\xi_{xyc}$  (see Fig. 19). A restricted Poincaré map is defined from  $S_\theta \cap Z$  to  $S_\theta \cap Z$ , where  $Z = \{(\theta, q, \dot{q}) | y_1 = 0, y_{2c} = 0, \dot{y}_{2c} = 0, y_{3c} = 0, \dot{y}_{3c} = 0\}$  describes the zero dynamics manifold (see (25)) and  $S_\theta = \{(\theta, q, \dot{q}) | \theta = 0.4\theta_i + 0.6\theta_f\}$  is the Poincaré section. The key point is that the state of the robot can be parametrized by three independent variables  $x = [q_2, \dot{q}_2, \dot{\theta}]^T$  in  $S_\theta \cap Z$ . The reduction of the states dimension is explained as follows.

In  $S_\theta \cap Z$ , from the third line of (25), the definition of  $u$  in (19) and  $u_c$  in (27), we have:

$$Q = u^d(\theta) - M_1 q_2. \quad (32)$$

It shows  $Q$  is determined by  $\theta$ ,  $q_2$  and  $M_1$ . The second line of (25) and the definition of  $\xi_{xy}$  in (12), (13) show that  $q_{12}$  and  $q_{13}$  can be calculated by  $q_1, q_2, \dots, q_{11}$ . In fact, according to the definition of  $Q$  and  $\theta$  (see (20), (8) and (32)), we can deduce that  $q_{12}$  and  $q_{13}$  depends on  $q_2$ ,  $\theta$  and  $M_1$ . Thus the calculation of  $q_{12}$  and  $q_{13}$  can be denoted as a function  $f_q()$ :

$$[q_{12}, q_{13}] = f_q(q_2, \theta, M_1). \quad (33)$$

(20) and (8) show that there exists a linear relation between  $q$  and  $Q$ ,  $q_{12}, q_{13}$ ,  $\theta$ ,  $q_2$ :

$$q = M_h [Q, q_{12}, q_{13}, \theta, q_2]_{14 \times 1}^T \quad (34)$$

where  $M_h$  is a  $14 \times 14$  constant matrix.

Therefore, (34) shows the full state of the robot  $q$  is determined by  $q_2$ ,  $\theta$  and  $M_1$ . Consequently, a simplified expressions of  $q, \dot{q}$  is defined as:

$$[q, \dot{q}] = f_{HZD}(q_2, \theta, \dot{q}_2, \dot{\theta}, M_1, C_{q^d}) \quad (35)$$

where  $C_{q^d}$  denotes all the known terms such as the reference trajectory  $q^d$  and its derivatives.

In consequence the acceleration  $\ddot{q}$  can be expressed as function of  $q_2$ ,  $\theta$ ,  $\dot{q}_2$ ,  $\dot{\theta}$ ,  $\ddot{q}_2$ ,  $\ddot{\theta}$  and it depends on  $M_1$  and  $C_{q^d}$ . Thus (25) corresponds to:

$$\begin{cases} W_{HZD}(q, \dot{q}) \begin{bmatrix} \ddot{q}_2 \\ \ddot{\theta} \end{bmatrix} + H_{HZD}(q, \dot{q}) = \mathbf{0}_{2 \times 1} \\ [q, \dot{q}] = f_{HZD}(q_2, \theta, \dot{q}_2, \dot{\theta}, M_1, C_{q^d}) \end{cases} \quad (36)$$

where the first line of (36) is deduced by the ZMP controller (11).  $W_{HZD}(q, \dot{q})$  denotes all the coefficient terms of  $\ddot{q}_2$  and  $\ddot{\theta}$ , and  $H_{HZD}(q, \dot{q})$  denotes all the terms without these terms.

Obviously, with definition of  $HZD$ , the first line of (36) can replace the original resulted dynamic equation (30) to describe the behavior of the robot from the Poincaré section until the impact. Just before the impact the complete states of the robot are calculated by using the second line of (36). Then the model of impact is used and  $u_c$ ,  $\xi_{xyc}$  can be calculated. The behaviour of the robot from the beginning of the next step to the Poincaré section is then calculated by equation of HZD that is similar to (36) but includes the modification terms of the reference trajectory  $u_c$ ,  $\xi_{xyc}$ .

### 5.2. Stability analysis at a fixed-point

The known cyclic motion  $q^d(\theta)$  gives a fixed-point  $x^{z*} = (q_2^d(\theta^*), \dot{q}_2^d(\theta^*), \dot{\theta}^*)$  for the proposed control law for any value of  $M_1$ , where  $\theta^* = 0.4\theta_i + 0.6\theta_f$ .

The restricted Poincaré map  $P^z : S_\theta \cap Z \rightarrow S_\theta \cap Z$  induces a discrete-time system  $x_{k+1} = P^z(x_k)$ . [29] states that, for  $\varepsilon$  sufficiently small in (28) and (29), *the linearization of  $P^z$  about a fixed-point determines exponential stability of the full order closed-loop robot model*. Define  $\delta x_k = x_k - x^*$ , the Poincaré map

linearized about the fixed-point  $x^*$  gives rise to a linearized system,

$$\delta x_{k+1} = A^z \delta x_k, \quad (37)$$

where the  $(3 \times 3)$  square matrix  $A^z$  is the Jacobian matrix of the Poincaré map, and it can be calculated as shown in [16]. A fixed-point of the restricted Poincaré map is locally exponentially stable, if, and only if, the eigenvalues of  $A^z$  have magnitude strictly less than one [15, Chap. 4].

## 6. Examples

(36) shows that the dynamic property of the system is influenced by the choice of  $M_1$ , exactly, the choice of joint angles to be controlled or the modification of reference motion (see (23)). One objective of this part is to study the influence of choice of  $M_1$  on the stability and to obtain some  $M_1$  to stabilize the walking. The other objective is to compare the robustness of our control law with that of a classical method.

### 6.1. The effect of different controlled partial joints on the walking stability

An exploration technique is used to illustrate the effect of the choice of  $M_1$ . First, arbitrary 9 components of  $M_1$  are fixed to zero, then the largest magnitude of the eigenvalues of  $A^z$  are drawn as function of the 10<sup>th</sup> component of  $M_1$ . If they are less than 1, the resulted walking is stable. The analysis results of  $M_1(j, 1)$ ,  $j = 6, 7$  are shown in Fig. 5, where the red circles denote the largest magnitude of eigenvalues of  $A^z$  is less than one, that means the stable cases exist. It also shows that the eigenvalues of  $A^z$  are large than 1 when  $M_1 = \mathbf{0}_{10 \times 1}$ , that means the walking is not stable. In fact, in this case the controlled joint angles are  $u = [q_1, q_4, \dots, q_{11}, q_{14}]^T$  (see (19)) and the reference motion  $Q^d$  is not modified by the tracking error of  $q_2$  (see (23)).

### 6.2. The robustness of the control law with $M_1(6, 1) = 5$ for the rigid ground model

According to Fig. 5, when we choose  $M_1(6, 1) = 5$ , the walking should be stable. In order to prove the validity and superiority of our control law, an

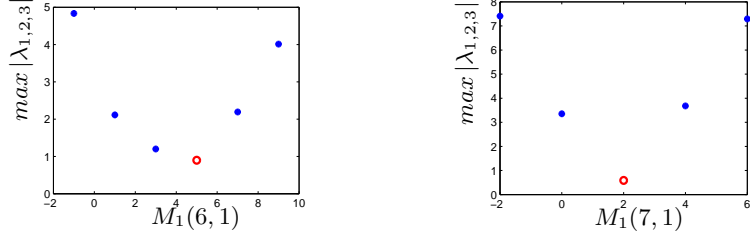


Figure 5:  $\max |\lambda_{1,2,3}|$  versus  $M_1(j, 1)$ ,  $j = 6, 7$ , when the other 9 components of  $M_1$  are zero.

example of this stable walking case is compared with a classical control law, in which all the joint angles are controlled and the desired properties of the closed loop system is:

$$\ddot{q} = \ddot{q}^d - K_d(\dot{q} - \dot{q}^d) - K_p(q - q^d), \quad (38)$$

where  $K_d > 0$  and  $K_p > 0$ . The joint torques can be obtained by (1) using  $q$ ,  $\dot{q}$  and  $\ddot{q}$ . Next, the same reference trajectory, the same simulator of the robot and the same simulation parameters are used for the two control laws. For the control law (38), in order to limit the influence of errors at the beginning of the step, the reference trajectory in (38) is also modified as presented in section 4.4.

Computation of impact and contact forces between foot and ground is an essential task for the simulation of a bipedal robot walking cycle. In our simulator, foot is supposed rigid and composed of four contact points  $P_1$ ,  $P_2$ ,  $P_3$  and  $P_4$  (see Fig. 2). If one of these points is in contact with the ground the corresponding reaction force is computed by solving a linear complementarity problem (LCP) [30]. Since the punctual contact at each corner of the foot  $P_1, \dots, P_4$  is considered, the slipping, rolling or take-off of the foot can be modeled. An initial error of  $0.01rad$  is introduced on  $q_2$  and a velocity error of  $0.01rad/s$  is introduced on  $\dot{q}_2$  and  $\dot{\theta}$ , the errors on the other joints are introduced to have a double support configuration.

The flat rigid ground model is used at first. For the proposed control law since the ZMP is controlled, the position of ZMP can track its desired value  $ZMP^d$  as shown in Fig. 6, thus the stance foot can remain flat on the ground

during single support phases. Moreover, it has been presented that the transferred foot is flat at each impact phases with the reference trajectory, that means the height of four points  $P_1$ ,  $P_2$ ,  $P_3$  and  $P_4$  of the swing foot are the same. Thanks to the swing ankle rotation controller, this condition is satisfied although not all the joint angles  $q$  are controlled to track  $q^d$  in the design of the overall control law, see Fig. 7. It describes the height of four points  $P_1, \dots, P_4$  for two feet respectively and it shows that the traces of  $P_1, \dots, P_4$  are almost the same, that means the supported foot remains flat on the ground and the swing foot is also flat before impact as prescribed for the reference motion, which can validate the swing ankle rotation controller. The walking motion of first three steps is shown in Fig. 11, it also displays the stable walking is achieved. The evolution of the error on  $q_2$  (the joint not directly controlled) is shown in Fig. 10. At the beginning the error increases to satisfy the condition on the ZMP but after that and step by step this error decreases and the cyclic desired motion is reached.

On the contrary, for the classical control law, since all the joint angles are controlled, the tracking error of the joint is close to zero at the end of the step (see Fig. 10, for the joint angle  $q_2$ ). However, the initial error is so high that the ZMP reaches the limits along the axis  $y$  as shown in Fig. 9. As a result, the stance foot rotates about its edges as displayed in Fig. 8. Fig. 12 also shows that the height of hips changes irregularly and the robot will fall down.

The comparison reconfirms that the control of ZMP is much more important than the tracking of predefined joints trajectory. It also indicates that the swing ankle rotation controller is helpful to the stable contact with the ground. In addition, the obtained stable walking shows that the choice of controlled partial joints using the stability analysis is effective.

### *6.3. The robustness of the control law with $M_1(6, 1) = 5$ for the soft ground model*

In order to further test the proposed control law, now a soft ground model is used in the simulator [31]. Obviously, it will propose higher requirements for the control law.

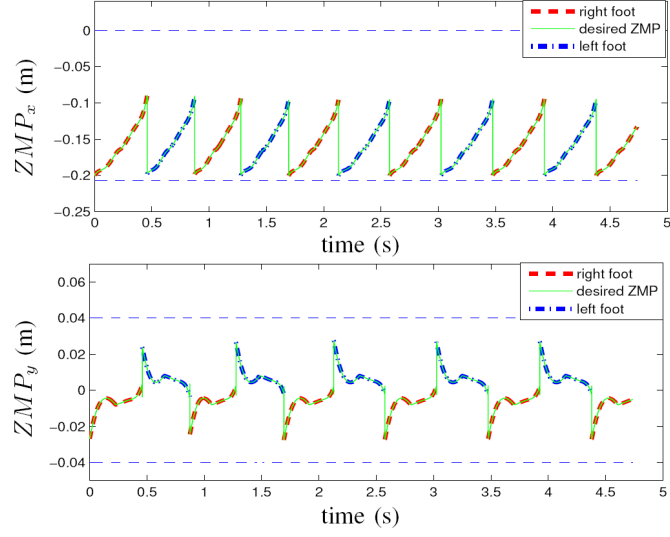


Figure 6: Position of ZMP in the foot sole with the proposed control law using the rigid ground model: it tracks its desired value and moves periodically.

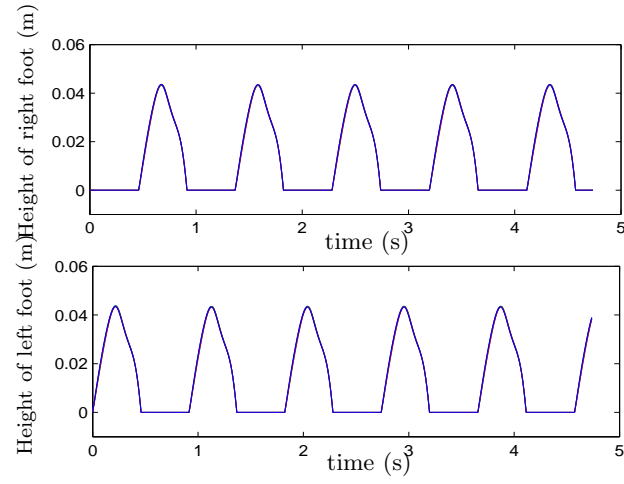


Figure 7: Height of feet with the proposed control law using the rigid ground model.

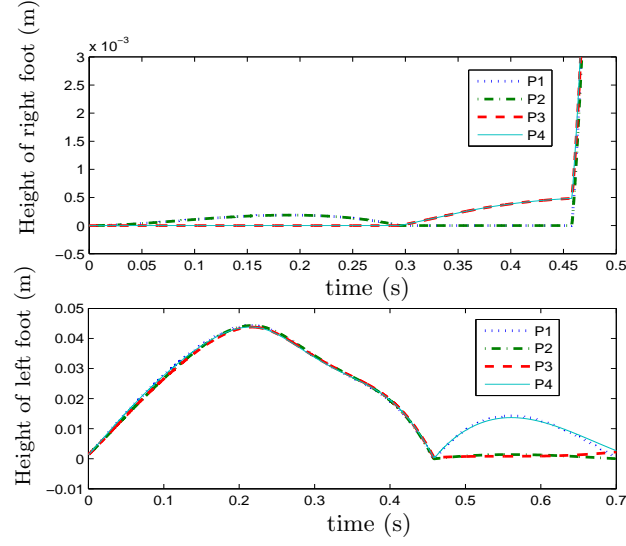


Figure 8: Height of feet with the classical control law, for the right foot a zoom is done along the vertical axis to show the unexpected rotation of the stance foot.

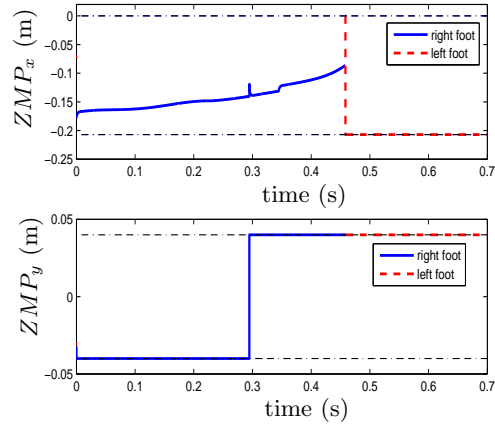


Figure 9: ZMP in the feet sole for the classical control law.

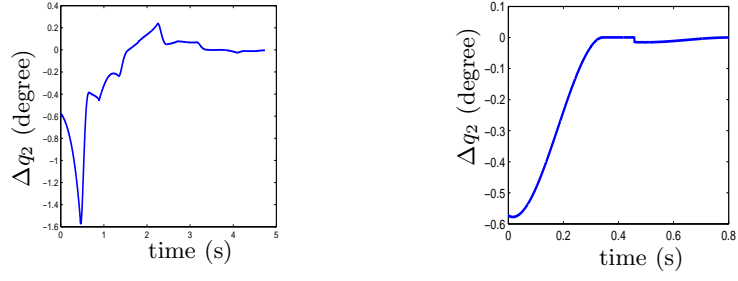


Figure 10: The tracking errors of  $q_2$  with an initial error for the proposed control law (left figure) and the classical method (right figure) for several steps.

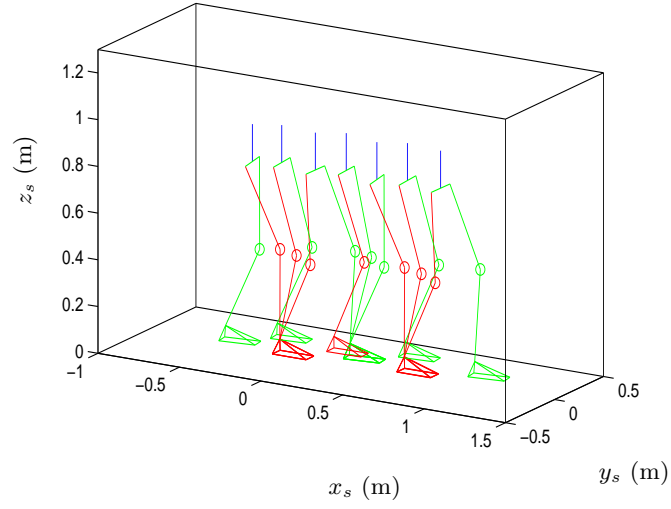


Figure 11: The walking motion of first three steps with the proposed control law.



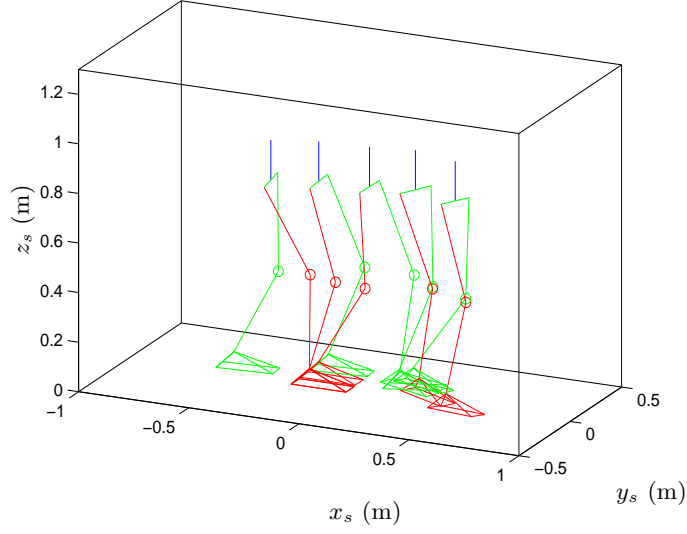


Figure 12: The walking motion of first two steps with the classical control law.

The same initial errors are introduced to every joint position and velocity as presented in the previous subsection. Since the soft ground model is used, the ground reaction force (GRF) changes acutely at the impact moment (see Fig. 13). As a result, ZMP is out of the polygon of support foot, but it is controlled quickly after the impact, as shown in Fig. 14. Consequently, the height of stance foot and the tracking error of  $q_2$  shake a little in the beginning of swing phase but at once recovers to zero (see Fig. 15 and Fig. 16). On the contrary, the walking is not stable with the classical control law. The results are similar to that using rigid ground model in Fig. 8 and Fig. 9 so they are not shown here.

## 7. Steering control

With the proposed control law, the robot can achieve an asymptotically stable and periodic walking along a straight line. It can be observed that the walking motion is not exactly along the  $x$ -axis with initial error. However, the robot is expected to be able to move all over the working place when it works in human environment. Thus we need the precise steering control not only the

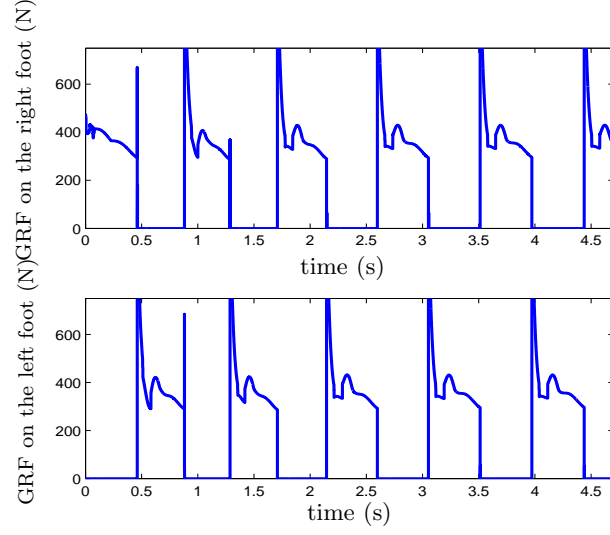


Figure 13: Ground reaction force (GRF) of the soft ground model.

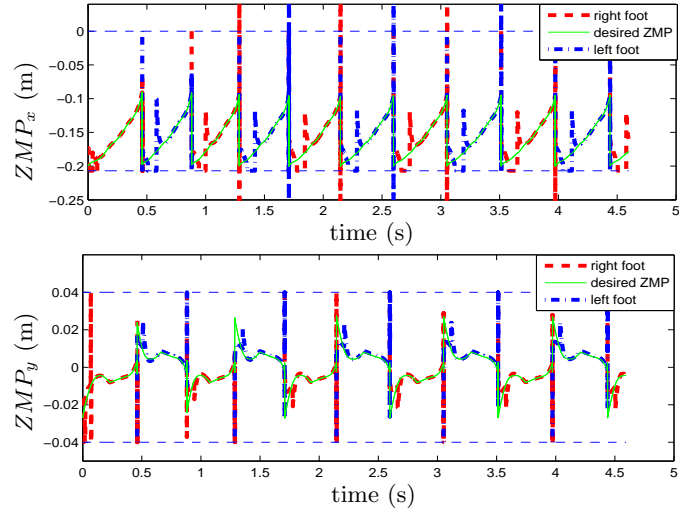


Figure 14: Position of ZMP in the feet sole with the proposed control law using the soft ground model: it moves periodically and returns to the limits fast after impact.

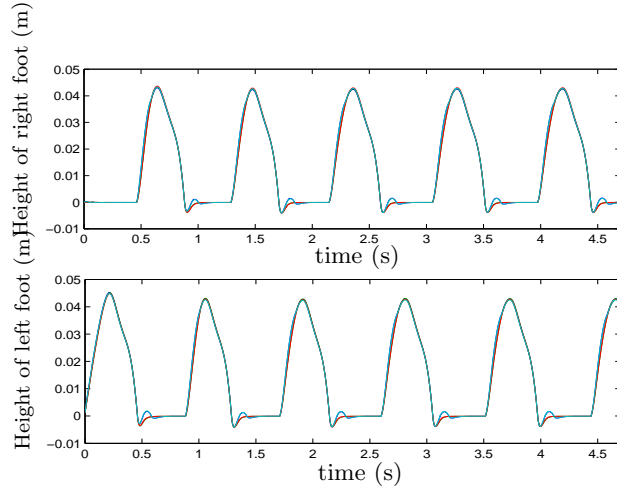


Figure 15: Height of feet with the proposed control law using the soft ground model.

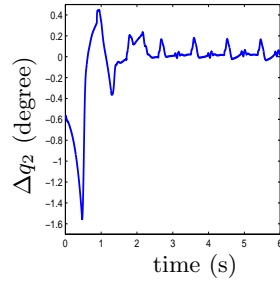


Figure 16: The tracking errors of  $q_2$  with the proposed control law using the soft ground model.

walking straight simply. Therefore, the objective of this section is adjusting the net yaw rotation of the robot over a step in order to steer the robot to walk along paths with mild curvature.

There are few studies about direction control. [32] and [33] studied the turning motions for biped with ZMP-based footstep planning. [34] proposed the robust direction control system that used gyro sensor feedback under environment with disturbance. In [35], motion stability during turning is ensured by adjusting the swing leg center of mass (COM) and hip position trajectories in a trial and error fashion. In this paper, an event-based feedback controller [36] is integrated with the proposed control law in Section 4 to regulate the walking direction through the net yaw motion about the stance foot over a step. Different from the previous studies, in our approach, the stability during steering is maintained. In addition, an interesting feature of this work is that one is able to control the robot's motion along a path with limited curvature using only a single predefined periodic motion.

#### 7.1. Stability analysis of the yaw motion

In order to describe the yaw motion of the robot, the walking direction angle  $q_0$  is supplemented to the configuration variables of the robot, that is:  $q^e = [q^T, q_0]^T$ . As shown in Fig. 17,  $q_l$  and  $q_r$  are respectively defined as the direction angle relative to  $x_s$ -axis of left foot and right foot respectively, the walking direction angle of the whole body  $q_0$  is given by:

$$q_0 = \frac{q_l + q_r}{2}. \quad (39)$$

Where  $q_l$  and  $q_r$  can be obtained using the position of two feet in the  $x_s y_s$  plane of the absolute frame.

Next, the stability of the extended system can be studied as shown in subsection 5.2. With the proposed control law, the behavior of the robot can be studied in the HZD. The extended states  $x^e = [q_2, \dot{q}_2, \dot{\theta}, q_0]^T$  is used and the corresponding linearized extended restricted Poincaré map at the fixed-point based on (37) is written as:

$$\delta x_{k+1}^e = A^e \delta x_k^e, \quad (40)$$

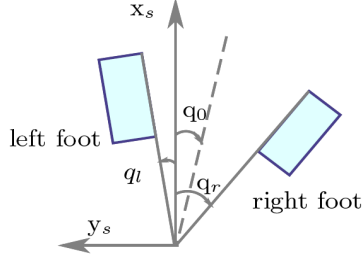


Figure 17: Description of walking direction

where  $A^e$  is a  $4 \times 4$  matrix and  $\delta x_k^e = x_k^e - x^{e*}$ . According to the linearization method of the Poincaré map about the fixed-point [16],  $A^e$  can be described as:

$$A^e = \begin{bmatrix} A_{\delta x_k, x_{k+1}}^e & A_{\delta q_{0k}, x_{k+1}}^e \\ A_{\delta x_k, q_{0k+1}}^e & A_{\delta q_{0k}, q_{0k+1}}^e \end{bmatrix} \quad (41)$$

where the four components of  $A^e$  denote the effect of introduced errors  $\delta x_k$  or  $\delta q_{0k}$  on the states of the next step  $x_{k+1}$  or  $q_{0k+1}$  respectively. Obviously,  $A_{\delta x_k, x_{k+1}}^e = A^z$ , which has been obtained in subsection 5.2. Moreover, the impact surfaces and the dynamic model are invariant under the rotation around the  $z_s$ -axis of the absolute frame [37]. Therefore, during the complete walking phase, a perturbation on  $q_0$  is conserved after one step and it doesn't affect the configuration of the biped. As a consequence the resulted  $A^e$  is rewritten as:

$$A^e = \begin{bmatrix} A^z & \mathbf{0}_{3 \times 1} \\ A_{\delta x_k, q_{0k+1}}^e & 1 \end{bmatrix} \quad (42)$$

Since the stability of the extended system is determined by the eigenvalues of  $A^e$ , in which the first three eigenvalues are the eigenvalues of  $A^z$  and the fourth eigenvalue is 1, that means the extended system is not stable, exactly, the yaw motion is not stable so the walking direction angle  $q_0$  can not be controlled by the control law proposed in Section 4.

### 7.2. Stabilization of the yaw motion

In order to stabilize the yaw motion, all the eigenvalues of  $A^e$  in (42) must be less than 1. Here an event-based feedback control [36] is introduced to

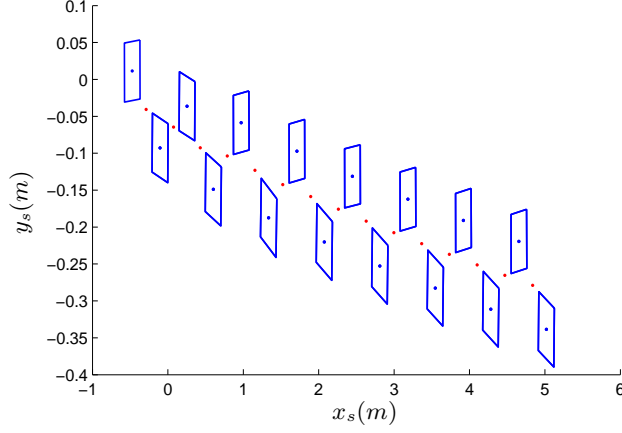


Figure 18: The positions of the feet at impacts with initial error of states, where the red points denote the midpoint of two feet. The robot departs the direction of  $x_s$  axis.

modify the the eigenvalues of  $A^e$ . Because the ZMP control and the swing ankle rotation control are always desired, so the event-based feedback control is only supplemented to the partial joint angles controller, then the desired property of the overall controller shown in (25) is rewritten as:

$$\begin{cases} y_1 = ZMP - ZMP^d(\theta) = 0 \\ y_{2c} = \xi_{xy} - \xi_{xyc} = 0 \\ y_{3cs} = u - u^d(\theta) - u_c - u_s = 0 \end{cases} \quad (43)$$

where the supplemented term  $u_s$  is defined as  $u_c$  in (26)

$$u_s = b_0 + b_1\theta + b_2\theta^2 + b_3\theta^3 \quad (44)$$

with  $b_0, b_1, b_2, b_3$  defined such that:

$$\begin{cases} u_s(\theta_f) = \beta \\ u_s(\theta) \equiv 0, \quad \theta \leq \theta_s \text{ and } \theta^* \leq \theta_s < \theta_f. \end{cases} \quad (45)$$

where  $\beta$  is the feedback control term to be determined, and it is retained constant during the swing phase and updated at each impact.  $\theta^*$  is the selected fixed-point in subsection 5.2 ( $\theta^* = 0.4\theta_i + 0.6\theta_f$ ).  $\theta_s$  denotes the introduction

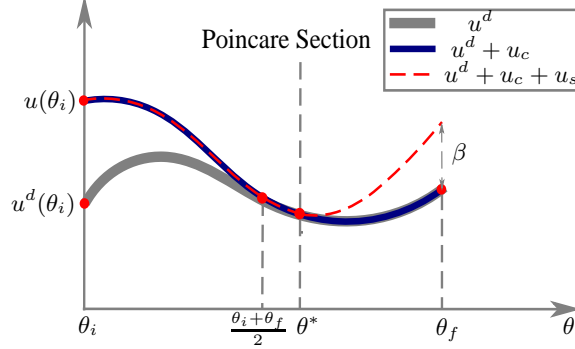


Figure 19: The modification of the reference path  $u^d$ .

moment of the event-based feedback control and it is limited by  $\theta^* \leq \theta_s < \theta_f$  to insure that the state of the robot is completely defined by  $x^e$  in the Poincaré section. In order to obtain sufficient time to control the walking direction before impact, here we choose  $\theta_s = \theta^*$ . By introducing  $u_c$  and  $u_s$  to  $u^d$ , the resulted reference trajectory can be described by Fig. 19. It should be noted that  $u_c$  modifies only the first part of the stance phase ( $u_c = 0$  when  $\theta \geq \frac{\theta_i + \theta_f}{2}$ ), and  $u_s$  modifies only the last part of the stance phase ( $u_s = 0$  when  $\theta \leq \theta_s$ ). Therefore, the choice of  $u_s$  in (45) is convenient because it does not require a re-design of the controller that created the HZD. The Poincaré map can now be viewed as a nonlinear control system on  $S_\theta \cap Z$  with inputs  $\beta_k$ , namely  $x_{k+1}^e = P^z(x_k^e, \beta_k)$ , where  $\beta_k$  is the value of  $\beta$  during the step  $k$ . Linearizing the Poincaré map about the fixed-point  $x^{e*}$  and the nominal parameter value  $\beta^* = \mathbf{0}_{10 \times 1}$  leads to

$$\delta x_{k+1}^e = A^e \delta x_k^e + F \delta \beta_k, \quad (46)$$

where  $\delta \beta_k = \beta_k - \beta^*$  and  $F$  is the Jacobian matrix of  $P^z$  with respect to  $\beta$ . Designing a feedback matrix

$$\delta \beta_k = -K \delta x_k^e \quad (47)$$

such that the eigenvalues of  $(A^z - FK)$  are strictly less than one will exponentially stabilize the fixed-point  $x^{e*}$ .

In (46),  $F$  is calculated numerically similar to the calculation of  $A^e$ . The

$(10 \times 4)$  gain matrix  $K$  is calculated via DLQR technique so that the state-feedback control law (47) minimizes the cost function  $\sum (\delta x_k^e \delta x_k^e + r \delta \beta_k' \delta \beta_k)$ . For  $r = 5$  the eigenvalues of the linearized Poincaré map in closed loop ( $A^e - FK$ ) are:

$$\begin{cases} \lambda_1 &= -0.3154 \\ \lambda_2 &= 0.1214 \\ \lambda_3 &= -0.0016 \\ \lambda_4 &= 0.0074 \end{cases} \quad (48)$$

All of them are less than 1 so the yaw motion will be stabilized and some examples of steering control are given in the next section.

### 7.3. Direction control of the yaw motion

It has been shown that the proposed feedback control law (47) can stabilize the system at the fixed-point  $x^{e*} = [q_2^*, \dot{q}_2^*, \dot{\theta}^*, q_0^*]^T$ . Moreover, as the model is invariant with respect to  $q_0$ , this control law can also stabilize the system at any fixed point given by  $x^{e*} + eC$  with  $e = [0, 0, 0, 1]^T$  and  $C$  a scalar constant. By consequence, the robot can converge to a motion with a desired direction of travel  $q_0^* + C$ . If  $C = 0$ , the walking direction angle converges to zero, otherwise, the direction of the robot is steered by changing  $C$ . With different expressions of  $C$ , different walking paths can be defined.

#### 7.3.1. Steering control of robot to walk along a desired direction

When the robot is desired to walk with a direction angle  $q_0^d$ , at the  $k^{th}$  step,  $C$  is defined as:

$$C_k = q_0^d. \quad (49)$$

According to  $\delta x_k^e = x_k^e - x^{e*}$ , there is:

$$\delta q_{0k} = q_{0k} - (q_0^* + C_k). \quad (50)$$

Here  $(q_0^* + C_k)$  can be regarded as the desired walking direction angle at the Poincaré section.



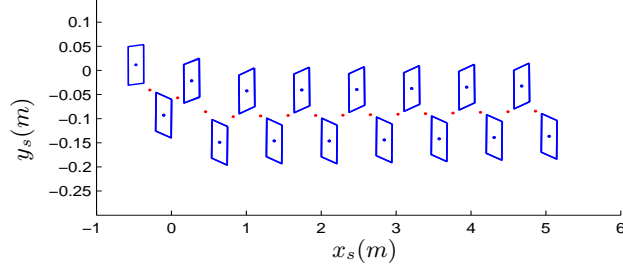


Figure 20: The positions of the feet at impacts under even-based feedback control, where the red points denote the midpoint of two feet. The robot walking along the direction of  $x_s$ -axis .

Considering the saturation of torques, the feedback term  $\delta q_{0k}$  used in (47) is chosen as:

$$\delta q_{0k-new} = \begin{cases} Q_{sat} & , \quad \delta q_{0k} > Q_{sat} \\ -Q_{sat} & , \quad \delta q_{0k} < -Q_{sat} \\ \delta q_{0k} & , \quad otherwise. \end{cases} \quad (51)$$

where  $Q_{sat}$  is a saturation of turning that must be chosen appropriately.

Finally, using (51) in (47), the even-based feedback control law can be calculated by:

$$\delta \beta_k = -K \begin{bmatrix} q_{2k} - q_{2}^* \\ \dot{q}_{2k} - \dot{q}_{2}^* \\ \dot{\theta} - \dot{\theta}^* \\ \delta q_{0k-new} \end{bmatrix}. \quad (52)$$

It should be noted that (52) only describes the feedback control law used in the case that the right leg is supported (see Fig. 1). All the work in this paper about the exchange of legs is based on a natural symmetry of the hybrid robot model [37]. When the supported leg is changed, the corresponding states of the other leg should be used in (52).

Here and in the follow examples we always choose  $Q_{sat} = 9^\circ$ . For  $C = 0$ , the simulation result is shown in Fig. 20. It indicates that with the same initial error as in Fig. 18, the walking comes back to the direction of  $x_s$ -axis by introducing the even-based feedback control law. In order to validate that the robot can

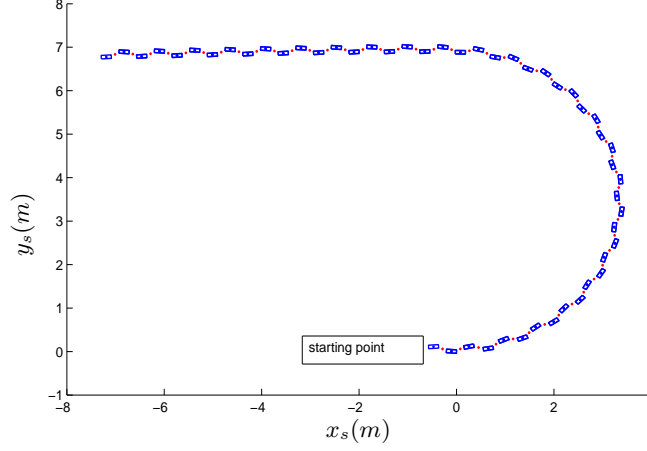


Figure 21: The positions of the feet at impacts under even-based feedback control, where the red points denote the midpoint of two feet. The robot walking along the direction of  $-x_s$ -axis .

turn with large extent, we choose  $C = 180^\circ$  and that means the robot will turn round and walking along the the direction of  $-x_s$ -axis . The resulted trajectory of the feet at impacts is shown in Fig. 21. Fig. 22 describes the direction angle of two feet  $q_l$ ,  $q_r$  and the walking direction angle  $q_0$ . It shows that  $q_0$  converges to  $180^\circ$  at last.

### 7.3.2. Steering control of robot to pass through a door

For some task, it can be desired that the robot pass through a point with a given orientation, as example to pass through a door. This case is illustrated as *Path 1* in Fig. 23. The control principle of *Path 1* is regulating the walking direction to eliminate the distance between the midpoint of two feet and the goal along  $y_s$ -axis , and then controlling the walking direction angle to 0. Therefore, at the  $k^{th}$  step,  $C$  is defined as:

$$C_k = k_1(y^d - y_k), \quad (53)$$

where  $y^d$  is the position of the door along  $y_s$ -axis and  $y_k$  is the current position of the midpoint of two feet at the  $k^{th}$  step.  $k_1$  is the control gain and  $0 < k_1 < 1$ . Substituting (53) in (50), and then using (51) and (52), the feedback control

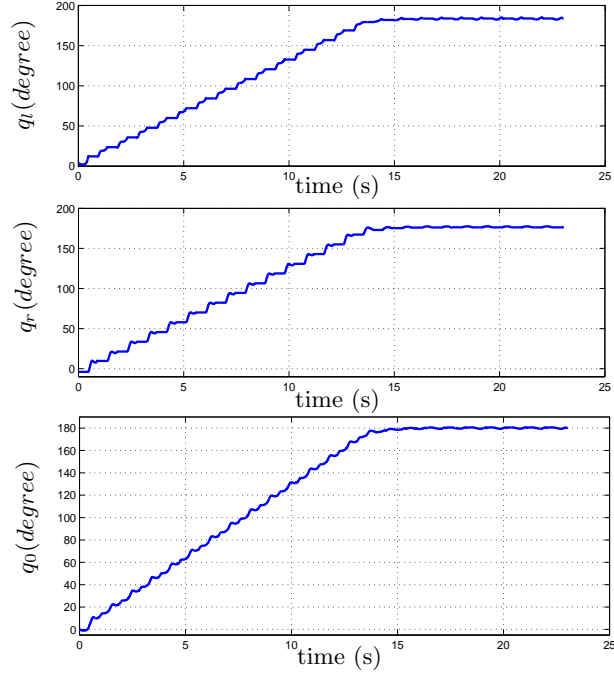


Figure 22: The direction angle of left foot  $q_l$ , right foot  $q_r$  and the walking direction angle  $q_0$ , where the straight lines in figures of  $q_l$  and  $q_r$  denote the single support phases.  $q_0$  converges to  $180^\circ$ .

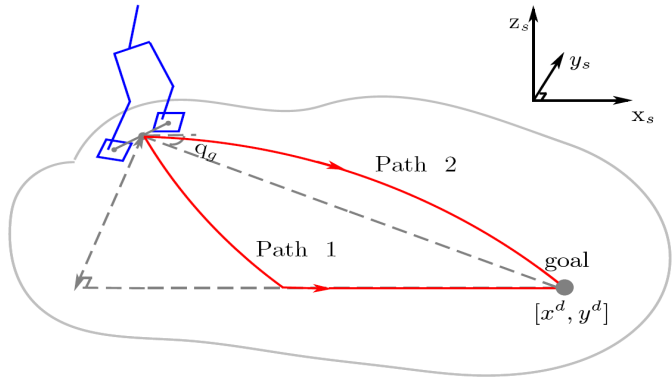


Figure 23: The layout of the robot and the goal.

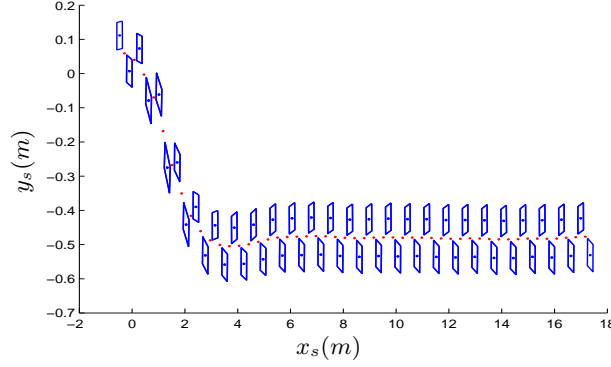


Figure 24: The positions of the feet at impacts for passing through the door, where the red points denote the midpoint of two feet. The walking path converges to the desired one  $y^d = -0.5$ .

law to let the robot pass through a door is obtained. Supposing  $y^d = -0.5$  and choosing  $k_1 = 0.8$ , the simulation results of *Path 1* are shown in Fig. 24 and Fig. 25. Fig. 24 indicates that the walking path converges to the desired one  $y^d = -0.5$ . Fig. 25 describes the direction angle of two feet  $q_l$ ,  $q_r$  and the walking direction angle  $q_0$ . It shows that  $q_0$  converges to 0 at last so the robot walks along the  $x_s$ -axis, that means it can pass through the door shown in Fig. 23.

#### 7.4. Steering control of robot to reach a destination

As shown in Fig. 23, if the robot is desired to reach a goal at the position  $[x^d, y^d]$ , the second path (*Path 2*) is more appropriate. Its objective is regulating the walking direction angle  $q_0$  to track the relative angle  $q_g$  between the robot and the goal in order that the walking direction of robot crosses the goal. The position of the midpoint of two feet in  $x_s y_s$  is  $[x, y]$ , at the  $k^{th}$  step  $q_g$  can be defined by:

$$q_{gk} = \begin{cases} \arctan(\frac{y^d - y_k}{x^d - x_k}) & , \quad x^d - x_k \geq 0 \\ \pi + \arctan(\frac{y^d - y_k}{x^d - x_k}) & , \quad otherwise. \end{cases} \quad (54)$$

According to the tracking objective,  $C$  is redefined to create a new fixed-point for the feedback even-based control law:

$$C_k = q_{gk}. \quad (55)$$

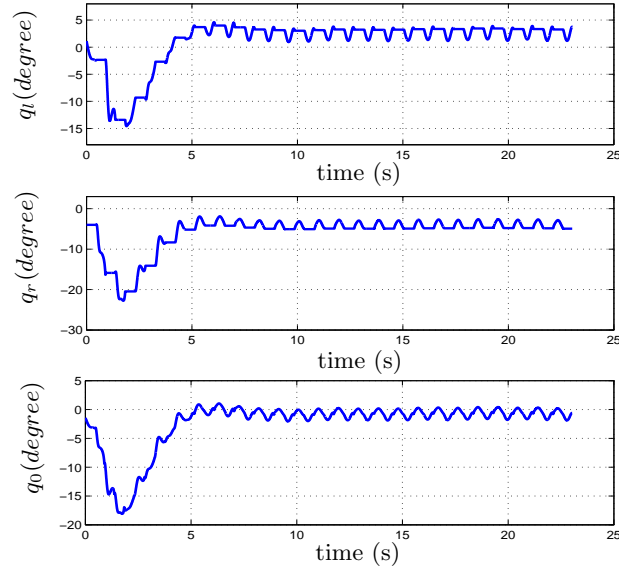


Figure 25: The direction angle of left foot  $q_l$ , right foot  $q_r$ , and the walking direction angle  $q_0$ , where the straight lines in figures of  $q_l$  and  $q_r$  denote the single support phases.  $q_0$  is regulated to obtain  $y^d = -0.5$  at first, then it converges to  $0^\circ$  so the robot walks along the  $x_s$ -axis as shown in Fig. 24.

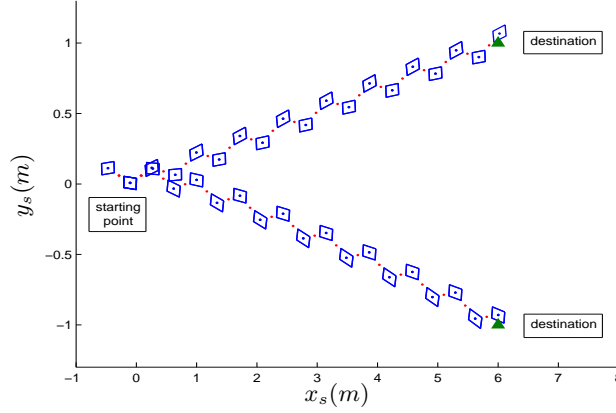


Figure 26: The positions of the feet at impacts for reaching two destinations at  $[6, 1]$  and  $[6, -1]$  respectively.

As soon as  $C_k$  is determined, using (50), (51) and (52), the feedback control law for the robot to reach a destination is obtained.

Two destinations at  $[6, -1]$  and  $[6, 1]$  in  $x_s y_s$  plane are chosen respectively for the simulation and Fig. 26 shows that the robot can reach these two points successfully.

## 8. Conclusions and perspectives

This paper proposed a walking control strategy for a 3D biped robot with flat-feet. Its objective is to control the ZMP, the swing ankle rotation and the partial joints angles simultaneously. By creating the hybrid zero dynamics, a stability study with application of Poincaré method was evaluated in a reduced space. The influence of the controlled partial joints selection on the stability of the control law was investigated. The examples showed that stability can be obtained by a pertinent choice of the controlled joints. Finally, a supplemental event-based feedback controller was designed to stabilize the yaw motion. By adjusting the set point of the event-based controller, it is possible to steer the direction of the robot, and even to direct the walking along a given path with mild curvature. In future, the proposed control law will be extended to the robot

model with the motion of arms and the foot rotation around the metatarsal axis.

### Appendix A. Computation of the angular acceleration and velocity of the joint 13 in the absolute reference frame

${}^s\omega_{13}$  and  ${}^s\dot{\omega}_{13}$  can be efficiently computed by the forward recursive equations of the general serial robot [20]. For the serial number of joints  $j = 1, \dots, 13$ , there exist:

$$\begin{cases} {}^j\omega_{j-1} = {}^jR_{j-1}{}^{j-1}\omega_{j-1} \\ {}^j\omega_j = {}^j\omega_{j-1} + \dot{q}_j {}^ja_j \\ {}^j\dot{\omega}_j = {}^jR_{j-1}{}^{j-1}\dot{\omega}_{j-1} + \ddot{q}_j {}^ja_j + {}^j\omega_{j-1} \times \dot{q}_j {}^ja_j \end{cases} \quad (\text{A.1})$$

where  ${}^ja_j = [0, 0, 1]^T$ ,  ${}^jR_{j-1}$  denotes the orientation matrix from the coordinate system  $x_j y_j z_j$  to  $x_{j-1} y_{j-1} z_{j-1}$ , and the initial condition are  ${}^0\omega_0 = 0$ ,  ${}^0\dot{\omega}_0 = 0$ . If  ${}^{13}\omega_{13}$  and  ${}^{13}\dot{\omega}_{13}$  have been computed by (A.1),  ${}^s\omega_{13}$  and  ${}^s\dot{\omega}_{13}$  in (14) can be obtained by:

$$\begin{cases} {}^s\omega_{13} = {}^sR_{13}{}^{13}\omega_{13} \\ {}^s\dot{\omega}_{13} = {}^sR_{13}{}^{13}\dot{\omega}_{13} \end{cases} \quad (\text{A.2})$$

### Appendix B. Acceleration constraint generated by the flat-foot impact condition

(16) can be rewritten as:

$${}^s\dot{\omega}_{13} {}^d_{xy} - {}^s\dot{\omega}_{13xy} + \frac{K_1}{\varepsilon} \dot{\xi}_{xy} + \frac{K_2}{\varepsilon^2} \xi_{xy} = \mathbf{0}_{2 \times 1}, \quad (\text{B.1})$$

where  $\xi_{xy}$  can be calculated by (12), (13) and  $\dot{\xi}_{xy}$  can be calculated by (A.1), (A.2) with  $q, \dot{q}$  and  $q^d, \dot{q}^d$ . According to Appendix Appendix A,  ${}^s\dot{\omega}_{13}$  is a linear function of  $\ddot{q}$ . Similar to (3), it can be defined as:

$${}^s\dot{\omega}_{13} = F_a \ddot{q} + F_v, \quad (\text{B.2})$$

where  $F_a$  and  $F_v$  can be obtained according to the functions to compute  ${}^s\dot{\omega}_{13}$ . Accordingly, as the same as (4) and (5), they can be calculated by:

$$F_v = {}^s\dot{\omega}_{13}(q, \dot{q}, \ddot{q} = \mathbf{0}_{14 \times 1}), \quad (\text{B.3})$$

and

$$\begin{cases} F_a(:, i) = {}^s\dot{\omega}_{13}(q, \dot{q}, \ddot{q} = \mathbf{e}_{14 \times 1}) - F_v \\ \mathbf{e}(i) = 1, \quad i \in \{1, 2, \dots, 14\} \\ \mathbf{e}(j) = 0, \quad j \in \{1, 2, \dots, 14\} \quad \text{and} \quad j \neq i \end{cases} \quad (\text{B.4})$$

According to (B.2), we can deduce  ${}^s\dot{\omega}_{13xy}$  and its desired value  ${}^s\dot{\omega}_{13}^d{}_{xy}$  are linear functions of  $\ddot{q}$  and  $\ddot{q}^d$  respectively. Since  $\ddot{q}^d$  includes a term about  $\ddot{\theta}$  (see (9)), which is also a linear function of  $\ddot{q}$  (see (8)). Therefore, (B.1) can be rewritten as:

$$W_s(q, \dot{q})\ddot{q} + H_s(q, \dot{q}) = \mathbf{0}_{2 \times 1}, \quad (\text{B.5})$$

where  $W_s(q, \dot{q})$  denotes all the coefficient terms of  $\ddot{q}$  and  $H_s(q, \dot{q})$  denotes all the terms without  $\ddot{q}$  in (B.1).

### Acknowledgments

This work is supported by ANR grants for the R2A2 project. Carlos Rengifo would like to acknowledge and express his sincere gratitude to *Universidad del Cauca* for the financial support given to him during this project.

### References

- [1] M. Vukobratovic, B. Borovac, D. Surla, D. Stokic, Biped Locomotion Dynamics, Stability, Control and Application, Springer-Verlag, Berlin, 1990.
- [2] T. Takenaka, T. Matsumoto, T. Yoshiike, Real time motion generation and control for biped robot-1st report: Walking gait pattern generation, in: Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009, pp. 1084–1091.
- [3] K. Hirai, M. Hirose, Y. Haikawa, T. Takenaka, The development of honda humanod robot, in: Proceedings of the IEEE International Conference on Robotics and Automation, Leuven, Belgium, 1998, pp. 1321 – 1326.
- [4] J. Kim, I. Park, J. Oh, Experimental realization of dynamic walking of the biped humanoid robot khr-2 using zero moment point feedback and inertial measurement, *Advanced Robotics* 20 (6) (2006) 707–736.



- [5] Q. Huang, K. Kaneko, K. Yokoi, S. Kajita, T. Kotoku, N. Koyachi, H. Arai, N. Imamura, K. Komoriya, K. Tanie, Balance control of a biped robot combining off-line pattern with real-time modification, in: Proceedings of International Conference on Robotics and Automation, San Francisco, USA, 2000, pp. 3346–3352.
- [6] H. Takeuchi, Development of mel horse, in: Proceedings of International Conference on Robotics and Automation, Seoul, Korea, 2001, pp. 3165–3171.
- [7] T. Sugihara, Y. Nakamura, H. Inoue, Realtime humanoid motion generation through zmp manipulation based on inverted pendulum control, in: Proceedings of the IEEE International Conference on Robotics and Automation, 2002, pp. 1404–1409.
- [8] J. Ferreira, M. Crisostomo, A. Coimbra, Neuro-fuzzy zmp control of a biped robot, in: Proceedings of the 6th WSEAS International Conference on Simulation, Modeling and Optimization, Lisbon, Portugal, 2006, pp. 331–337.
- [9] S. Coros, P. Beaudoin, M. van de Panne, Generalized biped walking control, in: Proceedings of the 37th International Conference on computer graphics and interactive techniques, Los Angeles, USA, 2010, to be appeared.
- [10] K. Mitobe, G. Capi, Y. Nasu, Control of walking robots based on manipulation of the zero moment point, *Robotica* 18 (2000) 651–657.
- [11] S. Kajita, O. Matsumoto, M. Saigo, Real-time 3d walking pattern generation for a biped robot with telescopic legs, in: Proceedings of International Conference on Robotics and Automation, Seoul, Korea, 2001, pp. 2299–2306.
- [12] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, H. Hirukawa, Biped walking pattern generation by using preview control of

- zero-moment point, in: Proceedings of IEEE International Conference on Robotics and Automation, Vol. 2, Taipei, Taiwan, 2003, pp. 1620 – 1626.
- [13] S. Kajita, M. Morisawa, K. Harada, K. Kaneko, F. Kanehiro, K. Fujiwara, H. Hirukawa, Biped walking pattern generator allowing auxiliary zmp control, in: Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 2006, pp. 2993 – 2999.
  - [14] D. Tlalolini, C. Chevallereau, Y. Aoustin, Human-like walking: Optimal motion of a bipedal robot with toe-rotation motion, IEEE/ASME Transactions on Mechatronics 16 (2) (2011) 310–320.
  - [15] E. Westervelt, J. Grizzle, C. Chevallereau, J. Choi, B. Morris, Feedback Control of Dynamic Bipedal Robot Locomotion, Control and Automation, CRC Press, Boca Raton, 2007.
  - [16] C. Chevallereau, J. Grizzle, C. Shih, Asymptotically stable walking of a five-link underactuated 3d bipedal robot, IEEE Trans. on Robotics 25 (1) (2009) 37–50.
  - [17] T. Wang, C. Chevallereau, Stability analysis and time-varying walking control for an under-actuated planar biped robot, Robotics and Autonomous Systems 59 (6) (2011) 444–456.
  - [18] R. Gregg, M. Spong, Reduction-based control of three dimensional bipedal walking robots, International Journal of Robotics Research 29 (6) (2010) 680–702.
  - [19] C. Chevallereau, J. Grizzle, C. Shih, Steering of a 3d bipedal robot with an underactuated ankle, in: Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 2010, to be appeared.
  - [20] W. Khalil, E. Dombre, Modeling, identification and control of robots, Hermes Sciences, Europe, Hermes Sciences, Europe, Paris, 2002.

- [21] C. Chevallereau, G. Abba, Y. Aoustin, F. Plestan, E. Westervelt, et al., Rabbit: A testbed for advanced control theory, *IEEE Control Systems* 23 (5) (2003) 57–78.
- [22] F. Plestan, J. Grizzle, E. Westervelt, G. Abba, Stable walk-ing of a 7-dof biped robot, *IEEE Transactions on Robotics and Automation* 19 (4) (2003) 653–668.
- [23] E. Westervelt, J. Grizzle, D. Koditschek, Hybrid zero dynamics of planar biped walkers, *IEEE Transactions on Automatic Control* 48 (1) (2003) 42–56.
- [24] C. Chevallereau, D. Djoudi, J. Grizzle, Stable bipedal walking with foot rotation through direct regulation of the zero moment point, *IEEE Trans. on Robotics* 24 (32) (2008) 390–401.
- [25] D. Djoudi, C. Chevallereau, J. Grizzle, A path-following approach to stable bipedal walking and zero moment point regulation, in: *IEEE International Conference On Robotics And Automation*, 2007.
- [26] J. Luh, M. Walker, R. Paul, On line computational scheme for mechanical manipulators, *Trans. of ASME, J. of Dynamic Systems, Measurement and Control* 102 (2) (1980) 69–76.
- [27] H. Nijmeijer, A. van der Schaft, *Nonlinear Control Dynamical Systems*, Springer-Verlag, Berlin, 1989.
- [28] A. Isidori, *Nonlinear Control Systems*, Springer-Verlag, Berlin, 1995, third edition.
- [29] B. Morris, J. Grizzle, Hybrid invariant manifolds in systems with impulse effects with application to periodic locomotion in bipedal robots, *IEEE Transactions on Automatic Control* 54 (8) (2009) 1751–1764.
- [30] Acary, Vicent, Brogliato, Bernard, *Numerical Methods for Nonsmooth Dynamical Systems: Applications in Mechanics and Electronics*, Vol. 35 of

Lecture Notes in Applied and Computational Mechanics, Springer Verlag, 2008.

- [31] C. Rengifo, Y. Aoustin, C. Chevallereau, F. Plestan, A penalty-based approach for contact forces computation in bipedal robots, in: Proceedings of the 9th IEEE-RAS International Conference on Humanoid Robots, Paris, France, 2009, pp. 121–127.
- [32] K. Nishiwaki, S. Kagami, J. Kuffner, M. Inaba, H. Inoue, Online humanoid walking control system and a moving goal tracking experiment, in: Proceedings of the 2003 IEEE International Conference on Robotics and Automation, Taipei, Taiwan, 2003, pp. 911–916.
- [33] R. Kurazume, T. Hasegawa, K. Yoneda, The sway compensation trajectory for a biped robot, in: Proceedings of the 2003 IEEE International Conference on Robotics and Automation, Taipei, Taiwan, 2003, pp. 925–931.
- [34] K. Matsumoto, A. Kawamura, The direction control of a biped robot using gyro sensor feedback, in: Proceedings of the 11th IEEE International Workshop on Advanced Motion Control, Nagaoka, Niigata, 2010, pp. 137–142.
- [35] M. Yagi, V. Lumelsky, Synthesis of turning pattern trajectories for a biped robot in a scene with obstacles, in: Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2000, pp. 1161–1166.
- [36] J. Grizzle, Remarks on event-based stabilization of periodic orbits in systems with impulse effects, in: Proceedings of the Second International Symposium on Communications, Control and Signal Processing, 2003.
- [37] M. Spong, F. Bullo, Controlled symmetries and passive walking, IEEE Transactions on Automatic Control 50 (7) (2005) 1025–1031.